Oriented pooling for dense and non-dense rotation-invariant features

Wan-Lei Zhao, Hervé Jégou, Guillaume Gravier

Oriented pooling for dense and non-dense rotation-invariant features

Wan-Lei Zhao[1]
http://www.cs.cityu.edu.hk/~wzhao2
Hervé Jégou[1]
http://people.rennes.inria.fr/Herve.Jegou
Guillaume Gravier[2]
http://people.irisa.fr/Guillaume.Gravier

[1] INRIA
Rennes, France
[2] CNRS/IRISA
Rennes, France

Abstract

This paper proposes a pooling strategy for local descriptors to produce a vector representation that is orientation-invariant yet implicitly incorporates the relative angles between features measured by their dominant orientation. This pooling is associated with a similarity metric that ensures that all the features have undergone a comparable rotation. This approach is especially effective when combined with *dense oriented* features, in contrast to existing methods that either rely on oriented features extracted on key points or on non-oriented dense features. The interest of our approach in a retrieval scenario is demonstrated on popular benchmarks comprising up to 1 million database images.

1 Introduction

In the last decade, visual recognition has witnessed a sequel of major breakthroughs, most of them stemming from the introduction of local descriptors such as scale invariant features transforms (SIFT) [13, 20]. In particular, the bag-of-words representation [4, 22] has drastically modified how images are indexed by casting sets of local features into a vector representation. For image search, inverted files [22, 23] efficiently implement an effective similarity measure to compare images, especially for large vocabularies [15, 18], while preserving to some extent the desirable properties of local descriptors such as invariance to changes in scale and orientation. In image categorization, the vector representation underpinning Bag-of-words is well adapted to subsequent powerful machine learning techniques such as support vector machines (SVMs) [21].

Bag-of-words has been extended in various ways. This paper is mostly connected to one of this improvement, called *weak geometry consistency* WGC. It enriches the representation of each descriptor with the (quantized) characteristic scale and dominant orientation [7] associated with the region of interest. This additional information is exploited by a Hough-like voting procedure [5] to favor the images that have been scaled and rotated consistently.

More recently, alternative coding techniques have been proposed for local descriptors, such as the Fisher Vector [3, 16, 17] or VLAD [9]. In large-scale image retrieval, after dimensionality reduction and compression [9], the database images are represented by as

few as dozens bytes, thereby allowing the efficient search in hundred millions of images. These coding techniques have also exhibited superior performance in classification [3].

One of the merit of these new encoding approaches is that they usually rely on small visual vocabularies. This property is especially interesting for classification. In this context, densely extracting a large number of descriptors on a regular grid provides superior recognition performance, even though these dense features are not rotation-invariant. The quantization cost is small compared to that of bag-of-words relying on a large vocabulary, and compensates the cost of extracting more features. The resulting vectors are not sparse and therefore not indexed by inverted files, yet in image search competitive search timings are achieved by using alternative compression-based indexing strategies [9].

This paper makes the following contributions over the existing pooling approaches. First, we depart from most existing works by using *rotation-invariant dense features*. In image search, most systems rely on key-points or region detectors. A description relying on regular dense features achieves good performance but at the cost of loosing invariance to orientation, which is not desirable in many applications. Pooling on dominant orientations of the local features have been explored in the context of object classification [10, 11], however they do not enforce the relative orientation to be preserved. As we will see, using dense oriented features without a proper pooling strategy is not sufficient by itself: ignoring the relative orientation of patches introduces too much invariance. In our case, we aim at obtaining both the discriminative power conveyed by dense descriptors and invariance to orientation.

Our main contribution achieves this property. It is a novel pooling technique inherited from VLAD, which uses the dominant angle as a pooling variable, which is obtained as a byproduct from our oriented-invariant dense descriptor extraction. This pooling variable is combined with the quantization index. Our pooling approach is associated with a similarity measure that implicitly selects the relative orientation between images, similar to WGC. Our method departs from WGC by achieving this covariant property at the pooling level, while WGC relies on an additional orientation information provided per feature point. Our method produces a vector representation that is compatible with dimensionality reduction.

Our method is not without drawbacks. Similar to spatial pyramid [12], using dominant orientation as a pooling variable increases the dimensionality of the vector, typically by a factor 8 when quantizing the dominant orientation in 8 bins. However, our approach is still significantly better than the corresponding baseline at a fixed vector dimensionality, *i.e.*, when using a smaller vocabulary for SIFTs in our method. In most cases, this conclusion still holds after dimensionality reduction to a fixed vector size. Another point is that our similarity computation strategy requires to produce more distances between two input vectors in order to detect the relative orientation maximizing the similarity. This last point is partially alleviated by the use of Fourier domain.

Overall, our approach is interesting in many situations. It remains tractable for millions of images, and gives a significant improvement when applied either to SIFTs extracted from key-points or extracted on dense grid. It magnifies the interest of orientation-invariant dense features for image retrieval, leading to achieve the best performance ever reported with a vector image representation on INRIA Holidays [7] and the object recognition benchmark of University of Kentucky. The feature vector is compatible with dimensionality reduction, and provides an efficient and effective retrieval system with a short image representation.

Our paper is organized as follows. Section 2 briefly introduces VLAD as well as an improved version that will serve as our baseline for vector image representation. Section 3 describes our covariant pooling technique, which is subsequently evaluated in section 4.

2 Background: VLAD and the improved VLAD* baseline

The vector of locally aggregated descriptors (VLAD) [9] is an encoding technique that produces a fixed-length vector representation v from a set $\mathcal{X} = \{x_1, ..., x_m\}$ of m local d-dimensional descriptors (*e.g.*, SIFT, $d = 128$), which have been extracted from a given image. The VLAD computation procedure relies on a visual vocabulary $\mathcal{C} = \{c_1, \cdots, c_k\}$ where the dictionary (size k) is trained offline with k-means algorithm. It is used by a quantization function $q : \mathbb{R}^d \to \mathcal{C}$ that associates x_i to its Euclidean nearest neighbor in the vocabulary $\mathcal{C}$, as $q(x) = \arg\min_{c \in \mathcal{C}} \|x - c\|$. VLAD is a $d \times k$ vector, where each component is indexed by both the indices i and j associated to the quantization indexes and sift components, respectively. A component of VLAD vector $v = [v_{1,1}, \ldots, v_{i,j}, \ldots, v_{k,d}]$ associated with $\mathcal{X}$ is obtained as

$$v_{i,j} = \sum_{x \in \mathcal{X} : q(x) = c_i} x_j - c_{i,j}, \tag{1}$$

where x_j and $c_{i,j}$ are the j^{th} components of descriptor x and visual word c_i, respectively. As a post-processing, the vector v is ℓ_2-normalized.

VLAD* baseline. To boost the performance of VLAD, we use several recent pre- and post-processing operations to boost the accuracy of the original VLAD design.

1. We use the RootSIFT variant [1], since it always leads to better performances in the retrieval task. This simply amounts to square-rooting the (positive) components of the SIFT on output of the description software.

2. We apply the power-law normalization introduced in [17] for Fisher vector. It updates the individual components of the VLAD descriptor as

$$v_{i,j} := \mathsf{sign}(v_{i,j}) \times |v_{i,j}|^{\alpha}, \tag{2}$$

   where α is a constant in the range $(0, 1]$, which is fixed to $\alpha = 0.2$ in all our experiments. This processing is argued [9] to reduce the negative effect of visual bursts.

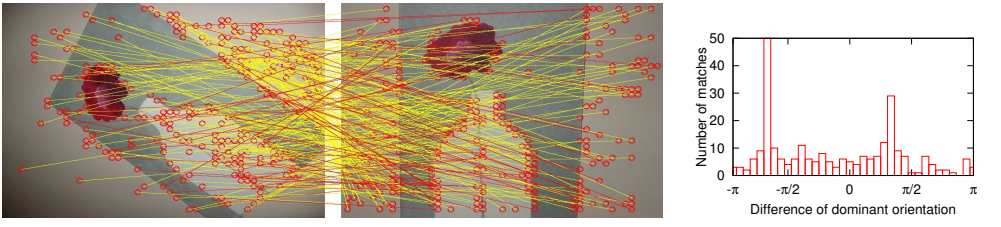
3. The power-law normalization is more effective if the input feature is rotated with PCA [9]. In our case, we do not reduce the dimensionality of features when performing this rotation (before power-law normalization), as we observe that dimensionality reduction is detrimental with VLAD, unlike for Fisher vectors for which dimensionality reduction is beneficial.

All these stages are applied prior to the ℓ_2-normalization, and gives an updated VLAD denoted by VLAD* in the rest of this paper.

Several other ameliorations have been proposed very recently, such as using multiple vocabularies to reduce the quantization noise [6] or introducing a per-cell normalization strategy instead of power-law [2]. We do not consider these complementary schemes in our paper, although we mention that they cover other aspects of VLAD and should be complementary with the approach introduced in our paper.

Oriented dense features. Most of the recent state-of-the-art papers on image classification and retrieval compute SIFT on regions of interest, or densely extract patches without considering orientation invariance. In contrast, we consider the interest of densely extracted

(a) HessianAffine



(b) Oriented Dense

Figure 1: Orientation-covariant features for two matching images. The lines connect the descriptors similar in the SIFT feature space. *Top:* Descriptors extracted in the Hessian-Affine detector [14]. *Bottom:* Descriptors densely extracted. In all cases, we estimate the orientation with the main gradient [13] and compute orientation covariant patches. The empirical histogram on the right gives the number of matching patches as a function of the angle difference between their dominant orientation. The lines in red correspond to the matches which leads to the majority of votes over all orientations. As one can observe, dense features produces more outliers, as to be expected because many patches on the border of the painting are similar. Yet the matching dense features cover the images more evenly, in contrast the ones associated with Hessian-Affine.

orientation-invariant features every 7 pixels[1] on the two canonical axes. The dominant angle is estimated with the main gradient [13]. The method introduced in our paper will avoid introducing too much invariance by incorporating the dominant angle at the pooling stage. Figure 1 illustrates the use of oriented features to facilitate the matching between two images where rotation is introduced.

3 Covariant pooling

This section introduces our main contribution, which is motivated by the observation that VLAD[2], does not control the degree of geometrical invariance. Either too much invariance is introduced when using orientation- or scale-invariant features, either these is not invariant if non-oriented features are densely extracted. This is in contrast with matching techniques such as WGC [7], which incorporate per-features geometrical information. Yet those are memory-demanding and are not vector representation suitable to dimensionality reduction to produce short vectors or codes as with VLAD [9].

[1] This step size is larger than the one commonly used in classification. This choice is probably sub-optimal, yet it limits the features at a reasonable size, compatible with real-time query processing.

[2] likewise Bag-of-features and the Fisher vector.

A key interpretation of VLAD is to view it as a cross-matching performed on aggregated features. In the regular VLAD, features with different orientation are aggregated, loosing the possibility to estimate any geometrical transformation between the aggregated features.

To alleviate this problem, we propose to pool features according to some characteristic geometrical quantities, more specifically the characteristic scales and dominant orientations [13] obtained as a byproduct of the descriptor computation stage. In other terms, we only aggregate features having similar characteristic scales or dominant orientations, to obtain a new pooling strategy termed Covariant-VLAD (CVLAD). Without loss of generality, the description of our method focuses on the dominant orientation. As we will see in the experimental section, it leads to the more effective pooling strategy.

Let denote by θ the dominant orientation associated with a given feature x, and let

$$b_B(\theta) = \left\lfloor B\frac{\theta}{2\pi} \right\rfloor \tag{3}$$

be the quantization function used to quantize angles with B equally sized bins. Our pooling strategy modifies Equation 1 as

$$p_{b,i,j} = \sum_{x\in\mathcal{X}:q(x)=c_i \wedge b_B(\theta)=b} x_j - c_{i,j}. \tag{4}$$

In Eqn. 4, the pooling of the feature x is controlled by both its quantization index $q(x)$ and its quantized dominant angle $b_B(\theta)$. Another way to see the CVLAD construction procedure is to consider that $\mathbf{P} = [\mathbf{P}_1, \ldots, \mathbf{P}_B]$ is a concatenation of B VLAD $k \times d$-dimensional vectors, each of which encodes the features having the same quantized dominant orientation. This produces a vector B times longer than VLAD. Similar pooling is feasible with the characteristic scale. Notice that the series of pre-processing and post-processing mentioned in section 2, in particular ℓ_2-normalization, are applied separately for each of the B VLAD sub-vectors. CVLAD is illustrated in Figure 2.

The similarity $s(.,.)$ between two CVLAD vectors $\mathbf{P}^i$ and $\mathbf{P}^j$ is defined on the basis of VLAD sub-vectors as

$$s(P^i, P^j) = \underset{\Delta t\in 0\ldots B-1}{\operatorname{argmax}} \sum_{t=0}^{B-1} \cos\left(\mathbf{P}^i_t, \mathbf{P}^j{}_{\bmod(t+\Delta t, B)}\right) \tag{5}$$

which amounts to selecting the orientation maximizing the similarity between the two vectors. This process is comparable to estimating the dominant rotation transformation between two feature sets in WGC [8], however here it is done directly on the aggregated vectors.

Improving matching efficiency with circulant encoding. Eqn. 5 performs a circulant matching between two sets of VLAD sub-vectors. The matching shifts the VLAD sub-vector in P^j circularly to search for the best match between two groups of VLAD. This introduces a complexity overhead when comparing the similarity metric of our CVLAD with that of VLAD vectors in the same size. More precisely, if done naively, the computation cost is multiplied by a factor B. Fortunately this computation overhead is partially alleviated by performing the circulant matching in the frequency domain, which is interesting for large values of B. This maximum correlation search strategy is common with temporal data, see for instance a recent paper on video matching [19].

Note that the method also allow us to restrict the comparison to a subset of possible rotations, possible only one. For instance, if no rotation is expected, Eqn. 5 becomes simply the direct cosine between the two CVLADs.

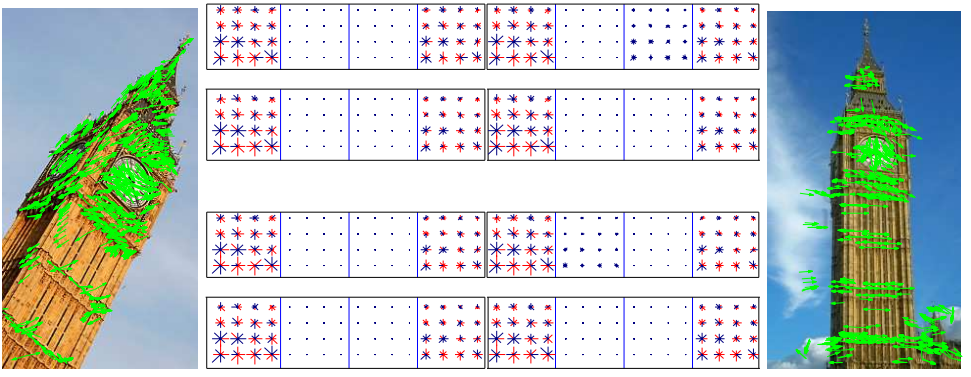


Figure 2: Illustration of the pooling scheme. Hessian-Affine features from each image have been quantized with four visual words and pooled in *8* orientations. Four sub-vectors of CVLAD (top two rows) from image on the left correspondingly match to four CVLAD sub-vectors from image on the right (bottom two rows). Their similarities are (in clockwise): *0.94*, *0.89*, *0.83* and *0.85*. Positive and negative components are colored in red and blue respectively.

Dimensionality reduction. One of the main interest of VLAD is to be a vector representation suitable to dimensionality reduction such as Principal component analysis (PCA). It is also true for CVLAD. In order to fit CVLAD in the context of large-scale task, CVLAD has to be mapped to lower dimension by PCA. However, the mapping should not treat CVLAD as whole, since if that, the pooling structure that is built in CVLAD will be destroyed. As a result, the PCA mapping is performed on each VLAD sub-vector. In order to enable the circulant matching between mapped sub-vectors, one universal mapping matrix is learned for all the sub-vectors. The sub-vectors are further ℓ_2-normalized right after the PCA mapping.

4 Experiments

This section evaluates the performance of VLAD with pooling (CVLAD) and compares it to existing approaches, Bag-of-words and regular VLAD. First we introduce the datasets used to analyze and evaluate our approach. Then we analyze the parameters and evaluates the interest of our approach with different feature detectors.

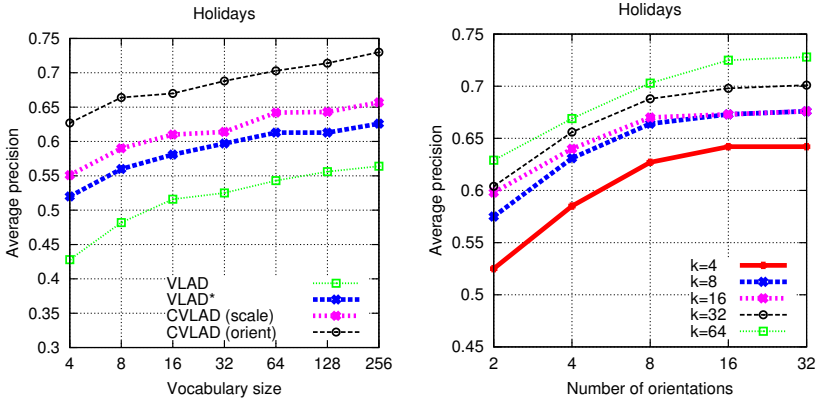
4.1 Datasets and evaluation protocol

The evaluation has been conducted with different settings and for three popular datasets, namely, Holidays dataset [8], the Oxford5k Building dataset [18] and the University of Kentucky object recognition benchmark [15]. Table 1 summarizes the main statistics and evaluation measures associated with each of this dataset. In our experiments, the visual vocabularies and PCA mapping matrices have been trained on a distinct image set.

Holidays [7] contains 1,492 images, which cover a large variety of scene types (natural, man-made, water and fire effects, etc) and images are of high resolution. 500 images have been selected as queries for each of the 500 partitioning groups on the image set.

Table 1: Statics on the four evaluation datasets

Dataset	Size	Num of queries	Performance Measure
Holidays [7]	1,492	500	mAP
Oxford5k [18]	5,063	55	mAP
UKB [15]	10,200	10,200	4×Recall@top4
Holidays + Flickr1M	1,001,492	500	mAP



(a) CVLAD *vs* VLAD and VLAD* (b) CVLAD with different orientation bins

Figure 3: Performances of pooling with dominant orientation (orient) and characteristics scale (scale). (a) In comparison with conventional VLAD and VLAD*. (b) Performances of pooling on orient with varying number of bins. VLAD, VLAD* and CVLAD are all computed with Hessian-Affine+SIFT.

Oxford5k contains images of Oxford famous buildings. There are 55 query images corresponding to 11 distinct buildings. A bounding box is provided for each query to define the area of the query image depicting the building.

University of Kentucky Benchmark (UKB) contains 2,550 different objects or scenes. Each one is represented by four images taken from four different viewpoints.

In addition to these three datasets, one million images collected from Flickr (the same as [8]) are adopted as distractors to evaluate the scalability of the proposed method.

4.2 Impact of the parameters

The effectiveness of VLAD* and CVLAD is first evaluated as a function of the vocabulary size ($k = 4, 8, \ldots, 256$). Observe in Figure 3(a) is that VLAD* consistently outperforms VLAD, which confirms the interest of the pre- and post-processing that we employ, see section 2 for details. These extra operations are adopted for CVLAD as well, which is compared to the stronger VLAD* baseline instead of the regular VLAD.

Figure 3(a) also shows the respective performance of CVLAD when constructed with either the dominant orientation or the characteristic scale associated with the local features. These quantities are pooled in $B = 8$ bins (in log-scale for characteristic scale), therefore the resulting CVLAD consists of 8 sub-vectors in both cases. Both variants of CVLAD (scale and orient) outperform the baseline. Pooling with the characteristic scale gives a performance on par with simply increasing the vocabulary size. Using the dominant orientation is noticeably better than using the scale and we therefore adopt it in the rest of our experiments.

Table 2: The Performances of the oriented pooling with *Hessian-Affine* and *Oriented dense* feature on Holidays, Oxford5k and Kentucky datasets.

k	Holidays (mAP)				Oxford5K (mAP)				Kentucky (Recall@top4)			
	HesAff		ODense		HesAff		ODense		HesAff		ODense	
	VLAD*	CVLAD	VLAD*	CVLAD	VLAD*	CVLAD	VLAD*	CVLAD	VLAD*	CVLAD	VLAD*	CVLAD
4	52.0	62.7	62.3	75.6	23.0	38.0	14.7	33.4	3.02	3.30	2.99	3.25
8	56.0	66.4	67.2	78.3	28.0	39.1	17.5	36.3	3.22	3.42	3.23	3.41
16	58.1	67.0	71.7	79.9	30.9	40.7	22.6	42.0	3.30	3.48	3.40	3.50
32	59.7	68.8	73.3	80.4	33.3	42.7	24.4	45.9	3.34	3.50	3.49	3.53
64	61.3	70.3	74.8	81.9	35.0	43.6	28.5	47.8	3.39	3.55	3.54	3.57
128	61.3	71.4	76.5	82.6	37.4	46.0	31.8	50.4	3.42	3.57	3.58	3.60
256	62.6	73.0	77.6	**82.7**	39.1	47.9	35.9	**51.4**	3.44	3.58	3.59	**3.62**

Considering vectors of fixed dimensionality, it is better to use orientation-based pooling than simply increasing the size of VLAD* vocabulary. For instance, with a given vocabulary size $k = 16$, CVLAD achieves mAP=67%, while VLAD* with $k = 128$ achieves mAP=61.3%. In the rest of our experiments, the pooling is only applied to dominant orientation and the number of orientations is fixed to 8, which as indicated in Figure 3(b), gives a good trade-off between performance gain and computational cost.

4.3 Oriented Dense Features

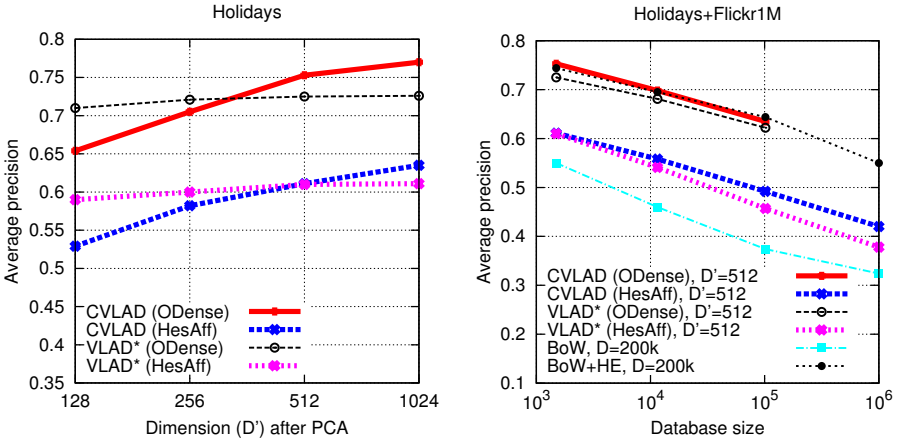
Table 2 gives the performance of CVLAD (based on orientation) on three popular benchmarks for varying vocabulary sizes. As a first rough observation, CVLAD achieves a performance similar to that of VLAD* with a vector size 4 to 8 times shorter.

Table 2 also analyzes the behavior of VLAD* and CVLAD-orient for two different feature detectors, namely *Hessian-Affine* (HesAff) and the *oriented dense* (ODense) extraction introduced in section 2. Unlike object classification task, dense sampling does not necessarily outperform the region detector if the features are aggregated with conventional VLAD. However, when the dense SIFT is coupled with oriented pooling, it is much better on Holidays and Oxford5K datasets. In particular, the mAP on Holidays dataset is already above mAP=80.0% with a small vocabulary, which is also quite competitive with the best results reported in [8]. The performances on UKB from different detectors are similar.

4.4 Large scale experiments

In order to make VLAD* and CVLAD more suitable for indexing large-scale databases, both of them are reduced to a lower dimensionality by PCA. For CVLAD, the mapping is performed on each VLAD sub-vector. CVLAD is built with vocabulary sized of $k = 16$. Each sub-vector is mapped to 64 dimensions. VLAD* is built with $k = 64$ visual words and reduced to 512 dimensions. CVLAD and VLAD* have therefore the same dimension after dimensionality reduction. The results of bag-of-words (BoW) and Hamming Embedding (BoW+HE) are also presented for reference. These approaches are significantly more costly in terms of both time and memory efficiency (by at least one order of magnitude).

Figure 4 shows that, with Hessian-Affine detector, the performances of VLAD* and CVLAD are similar if the reference set is small. However VLAD* suffers a faster performance drop than CVLAD as the reference set grows. The performance degradation of



(a) Performance of VLAD* and CVLAD after PCA

(b) Search in large-scale

Figure 4: Performance of CVLAD in large-scale image search as a function of database size. The performance of VLAD has been compared with CVLAD*, BoW and BoW+HE [8].

CVLAD* is similar to that of BoW+HE, but CVLAD has lower memory requirements. Oriented dense features boost the performances of CVLAD and VLAD* by more than 10%. CVLAD takes an edge on other pooling approaches, with a performance comparable to the one achieved by BoW+HE, and better than the BoW+WGC variant reported in [8].

5 Conclusion

We have presented a simple yet effective strategy, namely covariant pooling, to encode the descriptors based on some geometrical properties, and in particular their dominant orientations. This approach builds upon the recent VLAD descriptor, but offers a new trade-off with respect to geometrical invariance by implicitly selecting the rotation (likewise scale) to maximize the similarity for a given image pair.

Our CVLAD approach outperforms VLAD in almost all configurations at the cost of an increased comparison complexity. The pooling strategy increases the dimensionality, yet for a fixed one and/or after dimensionality reduction, CVLAD is shown of interest. When combined with dimensionality reduction, our approach is very efficient and easily scales to one million images in database.

Acknowledgements

This work was done as part of the Quaero project, funded by Oseo, the French agency for innovation.

References

[1] Relja Arandjelovic and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, Jun. 2012.

[2] Relja Arandjelovic and Andrew Zisserman. All about VLAD. In *CVPR*, 2013.

[3] Ken Chatfield, Victor Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[4] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *ECCV Workshop Statistical Learning in Computer Vision*, 2004.

[5] Paul V. C. Hough. Method and means for recognizing complex patterns. US patent 3069654, Dec. 1962.

[6] Hervé Jégou and Ondrej Chum. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *ECCV*, Oct. 2012.

[7] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, Oct. 2008.

[8] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87(3):316–336, Feb. 2010.

[9] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local descriptors into compact codes. In *Trans. PAMI*, Sep. 2012.

[10] Piotr Koniusz and Krystian Mikolajczyk. Spatial coordinate coding to reduce histogram representations, dominant angle and colour pyramid match. In *ICIP*, pages 661–664, Sept. 2011.

[11] Piotr Koniusz, Fei Yan, and Krystian Mikolajczyk. Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. *Computer Vision and Image Understanding*, 17(5):479–492, 2013.

[12] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, Jun. 2006.

[13] David Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2): 91–110, Nov. 2004.

[14] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, Oct. 2004.

[15] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, Jun. 2006.

[16] Florent Perronnin and Christopher R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, Jun. 2007.

[17] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, Sep. 2010.

[18] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, Jun. 2007.

[19] Jerome Revaud, Matthijs Douze, Cordelia Schmid, and Herve Jegou. Event retrieval in large video collections with circulant temporal encoding. In *CVPR*, 2013.

[20] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *Trans. PAMI*, 19(5):530–534, May 1997.

[21] Bernhard Schölkopf and Alexander Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

[22] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, Oct. 2003.

[23] Justin Zobel, Alistair Moffat, and Kotagiri Ramamohanarao. Inverted files versus signature files for text indexing. *ACM Trans. Database Systems*, 23(4):453–490, Dec. 1998.